Automated Plagiarism Detection Software System

Alapan Sau
International Institute of Information Technology, Hyderabad
alapan.sau@students.iiit.ac.in

***Abstract*—Plagiarism is one of the most critical forms of academic misconduct. It causes a negative impact on academic rigour. Plagiarism impedes the process of learning by distorting the process of evaluation. The course outcome is often harshly affected, and a sense of fake competition is created among the course participants. This paper proposes a solution based novel software system to detect plagiarism from various academic material comprising literature, mathematics and even source code of a software.**

***Index Terms*—Plagiarism, Academic Dishonesty**

I. Introduction

According to the academic policies of Oxford University[1], Plagiarism is presenting someone else's work or ideas as your own, with or without their consent, by incorporating it into your work without full acknowledgement. All published and unpublished material, whether in manuscript, printed or electronic form, is covered under this definition. Plagiarism may be intentional or reckless, or unintentional. Under the regulations for examinations, intentional or reckless plagiarism is a disciplinary offence.

Evaluation is an integral part of any education system. Evaluation helps to build an educational program, assess its achievements and improve upon its effectiveness. At an individual level of student, it helps to self-assess and work on improvement. However, Plagiarism is a plague to the Evaluation system. It corrupts the evaluation system and thereby breaks the quality and rigour of the academics.

In a survey of 24,000 students at 70 high schools, Donald McCabe (Rutgers University) found that 58% of students admitted to plagiarism and 95% said they participated in some form of cheating, whether it was on a test, plagiarism or copying homework[2]. A similar study by CollegeHumor found that from 30,000 college students, the percentage that cheated on their assignments was 60.8%, and 16.5% of them didn't regret doing it[3].

In an age of online referencing and free access to unlimited resources, it's easier than ever for students to plagiarise. There are thousands of websites which provides paid services involving assignments. In several parts of the world this academic dishonsty is now treated with higher concern.

We try to devise a software system that detects plagiarism in various academic material comprising the literature, mathematics and source code of software programs. The software system can be used by Course Instructors or Teaching Assistants across various universities invariant of the mode of education, language medium etc.

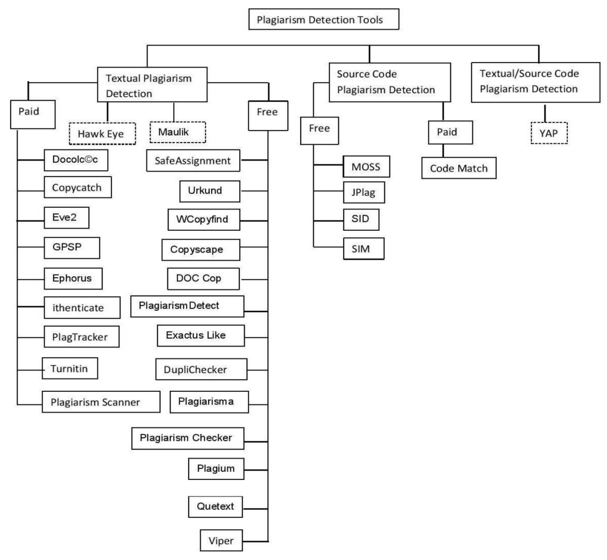

Fig. 1. Various Existing Plagiarism Detectors

II. Literature Review

There have been a wide range of approaches made in detection of plagiarism. These approaches can be classified based on the kind of plagiarism they detect - Source Code Plagiarism, Textual Plagiarism and Plagiarism in Mathematics. Chowdhury et al.[4] presented a very comprehensive list of available plagiarism detection systems.

A. Textual Plagiarism

The plagiarism detection in the literature can involve both textual and non-textual documents. Various Pre-processing techniques and Machine Learning Algorithms are proposed in detection. Some of the research highlights and existing tools in plagiarism detection are mentioned below. Chong[2] is one of the pioneers in using linguistic tools of Machine Learning in Plagiarism Detection. Natural Language Processing techniques were used instead of classical string matching in his research.

1) Plagiarisma: [5] It is free and simple plagiarism checking tool. This software supports 190+ languages and does not store any scanned content. The input file can be provided in three ways (1) Copy paste (2) Check by entering URL and (3) Uploading file. However, the tool lacks advanced features so it cannot be relied for heavy scanning work.

2) Viper: [6] This is also a free plagiarism detector that scans the submitted documents against 10 billion sources and documents present in a computer. This tool offers unlimited resubmitting of documents and it provides links to plagiarised work in the reports.

3) Turnitin: [7] This is another successful Web based tool provided by iPasradigms. The user is needed to upload a test document to the system database for plagiarism check. The system creates a fingerprint of the document and stores it. The toolcarries out the detection and report generation remotely. Turnitin is already accepted by 15,000 Institutions and 30 Million Students due to its simple user interface, support of large repository, rigorous text plagiarism check and well organized report generation. It can be considered as one of the best plagiarism checkers for the teachers.

4) Quetext: [8] This uses Natural Language Processing and Machine Learning to detect plagiarism. It first performs a internal plagiarism check. Then executes external checking. It provides support to multiple languages. It also has no word limit check. However one disadvantage of this tool is a disorganised report. It is also not very user friendly.

5) Hawk Eye: [9] It is an innovative plagiarism detection system. This uses mobile scanner OCR(Optical Character Recognition) engine into convert image to text and that text it uses as input. The OCR Engine pre-processes the clicked image in order to remove noise and disturbance from it and extract relevant keywords from image. The system uses plagiarism detection algorithms to remove unnecessary details like comments and changing variables names. or It uses string matching to detect plagiarism. It considers many limitations of existing well known plagiarism detection tools like Moss, JPlag, and Turnitin.

B. Source Code Plagiarism

In source code plagiarism, codes written by others are copied or reused or modified or converted a part of codes and claimed as one's own. A set of tools to determine software Plagiarism are available. Some of them are mentioned below

1) MOSS: [10] Moss is one of the the most popularly used source code plagiarism detection system. Aiken, from Stanford reports that the Moss web service has roughly 300K current Moss accounts, with 50K–100K new Moss accounts per year []. Moss is also intergrated with gradescope in recent times. Moss supports a total of 24 different programming languages.

2) JPlag: [11] It is a Web based source code plagiarism detection tool started in 1997. The tool accepts a set of programs as input to be compared and to present a report identifying matches. JPlag carries out programming language syntax and structure aware analysis to find results. It detects plagiarism in Java, C and C++ programs. The execution time of this service is quite fast(less than one minute for submissions of 100 programs of several hundred lines each.)

3) BPlag: [12] A modern technology of source code plagiarsm detection, inspired from ideals of MOSS, but with significantly better performances at many instances.

4) YAP3: [13] YAP is a system for detecting suspected plagiarism in computer program and other text submitted by the students. YAP3 is the third version of YAP which works in two phases. In the first phase, the source text is processed to generate token sequence. In second phase, each token is (non-redundantly) compared against all others strings.

5) SID-Software Integrity Diagnosis system: [14] It detects plagiarism between programs by computing the shared information. It uses a metric in measuring the amount of shared information between two computer programs, to enable plagiarism detection and the metric is approximated by a heuristic compression algorithm. SID works in two phases. In the first phase, source programs are parsed to generate token sequences by a standard lexical analyser. In the second phase, Token Compress algorithm is used to compute heuristically the shared information metric d(x; y) between each program pair within the submitted.

6) SIM: [15] This tool is to measure similarity between two C programs. It is useful for detection of plagiarism among a large set of homework programs. This tool is robust to common modifications such as name changes, reordering of statements and functions, and adding/removing comments and white spaces.

C. Plagiarism in Mathematics

A lot of academic work is very math intensive involving mathematical expressions and evaluations. However, not a lot of studies have been done in analysing mathematical content to Detect academic Plagiarism. One of the first studies in this domain is done by Norman Meuschke, Moritz Schubotz, Felix Hamborg,Tomas Skopal and Bela Gipp. They developed a first math-based detection approach by implementing and evaluating different feature comparison approaches using an open source parallel data processing pipeline built using the Apache Flink framework[16].

III. System Architecture

Here, we propose a novel software system which will be designed as an web application. The system architecture will consist of a frontend, backend and database.

A. Users

There are three main type of Users involved with the system:

1) Student: The students receive assignments,finish them, submit the assignment on the web application portal before deadline. They also receive grades after the evaluation of the assignments.

2) Teaching Assistant: The TAs create assignments, check status of submissions,change deadlines, evaluate assignments, run plagiarism detector, sets the penalty policies and finally grade the assignments

3) Course Instructor: Thee course Instructor can manage access permissions to TAs and students, check the grades of a student, check the assignments created by the TA, and view assignment submissions of students.

B. Work Flow

- The Course Instructor registers a course
- The Course Instructor provides respective limited access to the TAs and students.
- The TA creates an assignment. The TA sets a deadline for the created assignment. The TA also specifies the accepted submission file formats and directory structure for the submissions.
- The students gets a mail notifying the new assignment.
- The students view the assignment
- The students submit the assignment in required file format and directory structure. A subsequent script validates the assignment submission format.
- The students receives a confirmation mail regarding the submission
- The TA views the submission status which depicts the number of submissions and defaulters.
- The TA runs the plagiarism Detector on the submissions. Since this is a time consuming process, the TAs are notified with a mail on the completion of the Plagiarism Detection.
- The TA and the Course Instructor view the plagiarism detection reports.
- The Detector flags the plagiarised submissions. The TA sets a penalty policy for plagiarism, particular to the nature of the assignment and detector results.
- The TA evaluates the assignments and submit the grades of each student.
- The students gets a mail notifying the evaluation of assignments, their individual grades and penalties(if any).

C. Requirements

The system has various Functional and Non-functional Requirements. They are picted as follows:

1) Functional Requirements:

- **Login:** The users can login to the system.
- **Registering Course:** The Course Instructor registers a course to the system
- **Managing Access Permissions:** The Course Instructor gives access rights to the TA and students
- **Creating an Assignment:** The TA creates the assignments, sets the deadline, specifies the submission formats
- **Submitting Assignment:** The students submit the assignments, which gets successfully submitted only if the submission format is right.
- **Plagiarism Detection:** The TA/Course Coordinators runs Plagiarism Detection on the submissions. The Plagiarism Detector generates a report.
- **Viewing the Plagiarism Report:** The Course Instructors and the TA can view the detailed report of the Detector. The TA/Course Coordinators can sort/filter/search based on the Plagiarism Extent, Names of Students etc.
- **Setting Penalty Policy:** The TA should be able to set a penalty policy based on the Plagiarism Extent
- **Evaluate the Submissions:** The TA/Course Coordinators can evaluate the submissions and grade the students.

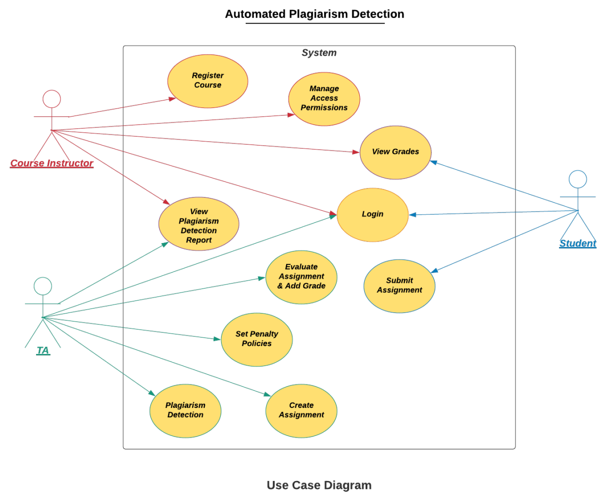

Fig. 2. UML Use Case Diagram

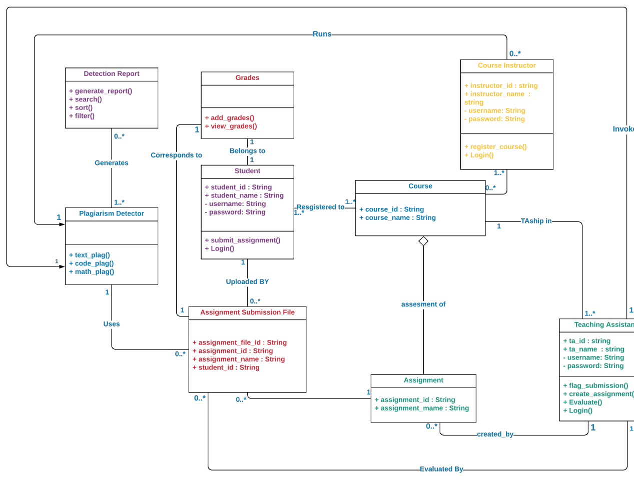

Fig. 3. UML Class Diagram

- **View Grades:** The students should be able to view their individual grades after evaluation.

2) Non-Functional Requirements:

- **Security:** The security of the application is of utmost importance.
- **Reliability:** Since the application can be accessed by a lot of people at the same time, it has to be very reliable.
- **Performance:** The performance of the website needs to be sufficiently fast, so that the last minute deadlines are not missed buffering!
- **Data Integrity:** The data integrity must be maintained across the database.
- **Usability:** The web application should be comprehensible and easy to use.

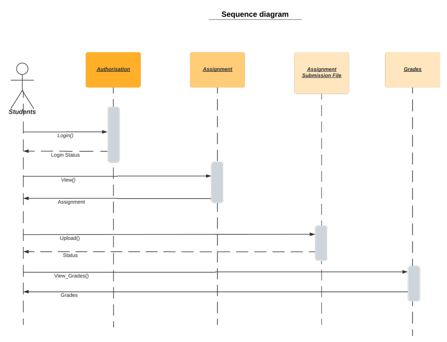

Fig. 4. UML Sequence Diagram (Students)

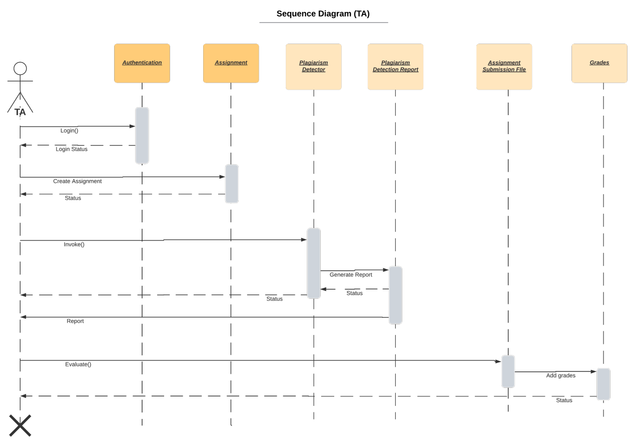

Fig. 5. UML Sequence Diagram (TA)

D. Classes

The following classes are used to build the system. These classes interact among themselves to support various functionalities. A brief description of the classes are given below:

*1) **Student:*** It represents the Student User which have a username and a password information to login.

*2) **Course Instructor:*** It represents the Course Instructor. This class has the responsibility of managing the access rights to the course for Student and TA

*3) **Course Instructor:*** It represents the Teaching Assistant of a course. It has the information of login credentials. It is responsible for creating new assignments, running Plagiarism Detector and evaluate Assignment.

*4) **Course:*** This class represents a course. All other classes are centred around this class

*5) **Assignment:*** This class represents a Assignment created by a TA in a particular course. The assignment class holds information regarding the assignment and is viewed by the student.

*6) **Assignment Submission File:*** This class represents the submission file of a student. These files are used by the Plagiarism Detector to detect plagiarism. The Files are also evaluated by the TA to add marks to the Gradesheet.

*7) **Gradesheet:*** This class holds the information regarding the grades of a student. A new grade is added every time, a TA evaluates an Assignment submission file.

*8) **Plagiarism Detector:*** This class represents the central aspect of this paper. The plagiarism detector is invoked by the TA or the course Instructor to generate the reports for all the assignment submission files.

*9) **Plagiarism Detection Report:*** The detection report holds information regarding the analysis made by the plagiarism detector. It handles the various sort, filter, search methods to get proper information in required from.

IV. Conclusion

The paper discusses a novel way to build a software system used for the purpose of Automated Plagiarism Detection. The paper proposes a wide range of possible combinations of plagiarism detection. If a student is found to violate the policies of academic honesty, the TA or the course instructor can simply detect is using this technology.

This solution however is not foolproof. The main issue with the given solution is that, it does not define a properly discrete Plagiarism Value to flag the student. The optimal point to determine whether or not to flag a student has to be decided manually by the TA or course coordinator, as this value will not be invariant to the Assignment, or written paper.

The world is moving towards a better future. However this increasing breach of academic honesty over time always brings up a lot of questions regarding the morals and values we are giving to our next generations.

The academic strictness, advancement in plagiarism detection technology is just a temporary fix. However the problem is quite deep underneath. After all we do not want a society where people are scared to unfollow rules, or indulge in act of dishonesty, rather we want a society which values a cooperation and honesty.

References

[1] Reference: https://www.ox.ac.uk/students/academic/guidance/skills/plagiarism
[2] Man Yan Miranda Chong. 2013. A study on plagiarism detection and plagiarism direction identification using natural language processing techniques. Ph. D Thesis. University of Wolverhampton.
[3] Taiseer Abdalla Elfadil Eisa, Naomie Salim, and Salha Alzahrani. 2015. Existing plagiarism detection techniques: A systematic mapping of the scholarly literature. Online Inf. Rev. 39, 3 (2015), 383–400.
[4] 4. Hussain A. Chowdhury and Dhruba K. Bhattacharyya. 2016. Plagiarism: Taxonomy, tools and detection techniques. In Proceedings of the 19th National Convention on Knowledge, Library and Information Networking (NACLIN'16).
[5] Reference: http://plagiarisma.net/
[6] R. R. Naik, M. B. Landge, C. N. Mahender, A review on plagiarism detection tools, International Journal of Computer Applications 125 (11).
[7] A. H. Osman, N. Salim, M. S. Binwahlan, Plagiarism detection using graphbased representation, arXiv preprint arXiv:1004.4449.
[8] Reference: http://www.quetext.com/
[9] P. Mulay, K. Puri, Hawk eye: Intelligent analysis of socio inspired cohorts for plagiarism, in: Innovations in Bio-Inspired Computing and Applications, Springer, 2016, pp. 29–42.
[10] Breanna Devore-McDonald, Emery D. Berger : "Mossad: Defeating Software Plagiarism Detection"

[11] L. Prechelt, G. Malpohl, M. Philippsen, Finding plagiarisms among a set of programs with jplag, J. UCS 8 (11) (2002) 1016.

[12] Hayden Cheers, Yuqing Lin, Shamus P. Smith: ”Academic Source Code Plagiarism Detection Measuring Program Behavioural Similarity”

[13] M. J. Wise, Yap3: Improved detection of similarities in computer program and other texts, ACM SIGCSE Bulletin 28 (1) (1996) 130–134.

[14] X. Chen, B. Francia, M. Li, B. Mckinnon, A. Seker, Shared information and program plagiarism detection, IEEE Transactions on Information Theory 50 (7) (2004) 1545-1551.

[15] D. Gitchell, N. Tran, Sim: a utility for detecting similarity in computer programs, in: ACM SIGCSE Bulletin, Vol. 31, ACM, 1999, pp. 266–270

[16] Norman Meuschke, Moritz Schubotz, Felix Hamborg, Tomas Skopal, Bela Gipp, ”Analyzing Mathematical Content to Detect Academic Plagiarism”